



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/605,794	06/27/2000	Osman Abdoul Ismael	P2651C	6721

7590 01/15/2002
William J Kubida Esq
Hogan & Harston LLP
1200 17th Street
Suite 1500
Denver, CO 80202

EXAMINER

FOURSON, GARY SCOTT

ART UNIT PAPER NUMBER

2151

DATE MAILED: 01/15/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

h

H-G

Office Action Summary

Application No. 09/605,794	Applicant(s) ISMAEL et al.
Examiner Gary Fourson	Art Unit 2151

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on Jun 27, 2000
- 2a) This action is FINAL.
- 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-17 is/are pending in the application.
- 4a) Of the above, claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-17 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claims _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are objected to by the Examiner.
- 11) The proposed drawing correction filed on _____ is: a) approved b) disapproved.
- 12) The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

- 13) Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).
 - a) All b) Some* c) None of:
 - 1. Certified copies of the priority documents have been received.
 - 2. Certified copies of the priority documents have been received in Application No. _____.
 - 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

*See the attached detailed Office action for a list of the certified copies not received.

- 14) Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

- 15) Notice of References Cited (PTO-892)
- 16) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 17) Information Disclosure Statement(s) (PTO-1449) Paper No(s). 6
- 18) Interview Summary (PTO-413) Paper No(s). _____
- 19) Notice of Informal Patent Application (PTO-152)
- 20) Other:

Art Unit: 2151

DETAILED ACTION

Claim Rejections - 35 U.S.C. § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. **Claims 1, 2, 7, 8, and 15-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hill et al. (US Patent 5,724,588) in view of Object Management Group (“The Common Object Request Broker: Architecture and Specification”).**

With respect to claim 17, at least one target object [Figure 3 object 301 and stub 302. Hill et al. teach stub 302 formation in Figure 8. Also see Figures 4A, 4B, 4C, 5, 12A, 12B, 14, and 15.], at least one network adaptor [The OMG teaches adaptors/ORBs for network connections in “The Common Object Request Broker: Architecture and Specification.” Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the ORB for network connections as taught by OMG, because Hill recognized that the Proxy-Stub connection for remote procedure calls allows for distributed processing, see column 5 lines 42-63.], the network adaptor(s) being registerable with the framework at the first machine [OMG page 32], and adaptor(s) being responsive to requests from client machine to the target object(s) [OMG page 8 teaches, “The Object Implementation information is provided at installation time and is stored in the Implementation Repository for use during request delivery.”].

With respect to claim 15, Hill et al. as modified by OMG for the rejection of claim 17 teaches the limitations substantially as claimed except the client application being able to both instantiate a client object and access/modify the target object(s) [Hill et

Art Unit: 2151

al., column 6 lines 63-64 state, “dynamically loading code to create an instance of the proxy, instantiating proxy 303, ...” It is notoriously well known that methods of an object may act upon or modify the encapsulated data of a target object in OOP applications.].

As to **claim 16**, Hill et al. as modified by OMG for claim 15 teaches the limitations substantially as claimed except the remote access support mechanism comprising a software mechanism [It was obvious to those of ordinary skill in the art at the time the invention was made that software mechanisms and hardware mechanisms are equivalent and interchangeable.].

With respect to claim 7, generating a client object forming a representation of a target object [Hill et al.: proxy 303 column 6 line 64], client object identifies and implements methods of the target object [Column 6 lines 66-67 state, “The client then accesses the interface of object 301 through proxy 303.” See also column 7 lines 1-3.], a network adaptor responsive to the client object [The OMG teaches adaptors/ORBs for network connections in “The Common Object Request Broker: Architecture and Specification.” Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the ORB for network connections as taught by OMG, because Hill recognized that the Proxy-Stub connection for remote procedure calls allows for distributed processing, see column 5 lines 42-63.], and the client object being configured to be instantiated by a client application enabling access as well as modification of the target object [column 6 lines 63-64 state, “dynamically loading code to create an instance of the proxy.” It is notoriously well known that methods of an object may act upon or modify the encapsulated data of a target object in OOP applications.].

As to **claim 8**, Hill et al. as modified by OMG for claim 7 teaches the limitations substantially as claimed except the client object being compiled from a target object [The

Art Unit: 2151

object oriented programming languages (i.e. C++, Smalltalk, and Java) are compiled into bytecodes for Java virtual machines or machine language for execution on processors.], the client object comprising a target object interface [See Hill et al. column 6 lines 33-34], and a target object stub implementing remotely accessible methods [Hill et al. state in column 9 lines 6-8, "In step 510, the stub invokes the method GetCell for the spreadsheet object passing the cell location."].

With respect to claim 1, Hill as modified by OMG for claim 7 teaches the limitations substantially as claimed except registering a target object [Hill et al. Teach in the ABSTRACT "the server process instantiates an object that has multiple interfaces." Hill et al. teach stub 302 formation in Figure 8, and block 803 shows registration of the stub.]and a network adapter [The OMG teaches adaptors/ORBs for network connections in "The Common Object Request Broker: Architecture and Specification". Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the ORB for network connections as taught by OMG, because Hill recognized that the Proxy-Stub connection for remote procedure calls allows for distributed processing, see column 5 lines 42-63.] for a network protocol [See Hill et al. column 19 lines 7-17.], associating the client object with a network adapter [The proxy objects of Figures 3, 4A, 4B, and 4C are associated with the network adaptor as the lines crossing the network boundary or CORBA middleware connect the proxy object with the corresponding target stub.], and enabling a client application to access the target object by instantiating the client object [Hill et al. Teach in the ABSTRACT, "instantiates a proxy object for receiving requests to invoke a function member of the interface and for sending the request to the identified stub."].

As to **claim 2**, Hill et al. as modified by OMG for claim 1 teaches the limitations substantially as claimed except the client object comprising a target object interface [See

Art Unit: 2151

Hill et al. column 6 lines 33-34] and a target object stub implementing the remote methods [col 7 lines 20-25].

3. Claims 4 and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hill et al. (US Patent 5,724,588) as modified by Object Management Group (“The Common Object Request Broker: Architecture and Specification”) as applied to claim 1 above, in view of Hughes (“JavaBeans and ActiveX go head to head”) and further in view of Hamilton et al. (US Patent 5,737,607).

As to **claim 4**, Hill et al. as modified by OMG for claim 1 teaches the limitations substantially as claimed except the client and target objects are beans [Hughes teaches compares JavaBeans with Microsoft’s ActiveX component frameworks. Distributed frameworks for the OOP component paradigm were routinely utilized at the time the invention was made. It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the JavaBean component framework for creation of the proxy and stub objects in conjunction with the object request broker, because Hamilton et al. teaches in column 2 lines 44-46, “It is therefore desired to allow Java programs to use different ORBs without requiring any changes to the Java program.”] and comprise properties methods and support for events and introspection [Hughes teaches JavaBeans support autodescription through an introspection mechanism on page 3. “The introspector class uses the BeanInfo class if it is supplied; otherwise, it uses the Reflection API.”].

As to **claim 5**, Hill et al. as modified by OMG for claim 4 teaches the limitations substantially as claimed except extracting target object methods by introspection [Hughes states on page 3, “With the Reflection API, one class can examine the methods provided by another class.”].

Art Unit: 2151

4. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hill et al. (US Patent 5,724,588) as modified by Object Management Group (“The Common Object Request Broker: Architecture and Specification”) as applied to claim 1 above, in further in view of Hollberg et al. (EP 0 727 739 A1).

As to **claim 6**, Hill et al. as modified by OMG for claim 1 teaches the limitations substantially as claimed except the target object is a managed object [Hollberg et al. teaches agent based management of managed objects. There is disclosed both ProxyAgent Objects in column 10, as well as Proxy Managed Objects in column 18 as local representatives of remote entities, see abstract. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the managed objects as disclosed by Hollberg et al. into the distributed CORBA based framework of Hill et al. as modified, because Hollberg recognized that “the object-oriented interface (OOI) for the use in OSI management applications and the related Object Interface Composer (OIC), minimize the effort needed to build the communication related functions of management applications.”] and the client application is a network management application [Hollberg et al. disclose an interface for converting network management application programs into network communication protocols, see abstract.].

5. Claims 3 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hill et al. (US Patent 5,724,588) as modified by Object Management Group (“The Common Object Request Broker: Architecture and Specification”) as applied to claims 1 and 7 above, in view of Stutz et al. (US Patent 5,517,645).

As to **claim 3**, Hill et al. as modified by OMG for claim 2 teaches the limitations substantially as claimed except selectively replacing the target object stub for dynamically modifying the behavior of said client application at runtime [col 21 lines 37 et seq. teaches replacement of a target object stub at runtime. Therefore, it would have

Art Unit: 2151

been obvious to one of ordinary skill in the art at the time the invention was made to selectively replace the target object stub at runtime as taught by Stutz et al., because Stutz et al. recognized that passivation of a stub in order to replace the connection to the remote object with the reverse interface stub is desirable in certain instances.].

As to **claim 9**, Hill et al. as modified by OMG for claim 8 teaches the limitations substantially as claimed except the target object stub being selectively replaceable for dynamically modifying the behavior of the client application at runtime [col 21 lines 37 et seq. teaches replacement of a target object stub at runtime. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to selectively replace the target object stub at runtime as taught by Stutz et al., because Stutz et al. recognized that passivation of a stub in order to replace the connection to the remote object with the reverse interface stub is desirable in certain instances.].

6. Claims 10 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hill et al. (US Patent 5,724,588) as modified by Object Management Group (“The Common Object Request Broker: Architecture and Specification”) and as further modified by Stutz et al. (US Patent 5,517,645) as applied to claim 9 above, in view of Hughes (“JavaBeans and ActiveX go head to head”) and further in view of Hamilton et al. (US Patent 5,737,607).

As to **claim 10**, Hill et al. as modified by OMG and as further modified by Stutz et al. for claim 9 teaches the limitations substantially as claimed except the client object being an object which comprises a set of properties, a set of methods for performing actions, and support for events and for introspection [Objects are known entities to provide encapsulation of data or properties accessible only by the encapsulating methods providing access to the data, and ‘get’ and ‘set’ methods are notoriously well known to provide basic manipulation in response to an outside event (i.e. another object calling the

Art Unit: 2151

method of the client object). Hughes teaches JavaBeans support autodescription through an introspection mechanism on page 3. “The introspector class uses the BeanInfo class if it is supplied; otherwise, it uses the Reflection API.” Distributed frameworks for the OOP component paradigm were routinely utilized at the time the invention was made. It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the JavaBean component framework for creation of the proxy and stub objects in conjunction with the object request broker, because Hamilton et al. teaches in column 2 lines 44-46, “It is therefore desired to allow Java programs to use different ORBs without requiring any changes to the Java program.”].

As to **claim 11**, Hill et al. as modified by OMG, Stutz et al., Hughes, and Hamilton et al. for claim 10 teaches the limitations substantially as claimed except the client object and the target objects are each beans which comprises a set of properties, a set of methods for performing actions, and support for events and for introspection [Hughes teaches compares JavaBeans with Microsoft’s ActiveX component frameworks. Hughes teaches JavaBeans support autodescription through an introspection mechanism on page 3. “The introspector class uses the BeanInfo class if it is supplied; otherwise, it uses the Reflection API.” Distributed frameworks for the OOP component paradigm were routinely utilized at the time the invention was made. It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the JavaBean component framework for creation of the proxy and stub objects in conjunction with the object request broker, because Hamilton et al. teaches in column 2 lines 44-46, “It is therefore desired to allow Java programs to use different ORBs without requiring any changes to the Java program.”].

7. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hill et al. (US Patent 5,724,588) as modified by Object Management Group (“The Common

Art Unit: 2151

Object Request Broker: Architecture and Specification”), Stutz et al. (US Patent 5,517,645), Hughes (“JavaBeans and ActiveX go head to head”), and Hamilton et al. (US Patent 5,737,607) as applied to claim 11 above, and further in view of Pelegri-Llopart et al. (US Patent 5,999,988)

As to **claim 12**, Hill et al. as modified by OMG, Stutz et al., Hughes, and Hamilton et al. for claim 11 teaches the limitations substantially as claimed except a compiler for extracting the target object methods by introspection for generating the client object [OMG teaches on page 32 that the interface is defined in Interface Definition Language and is used to generate the client Stubs and the object implementation skeletons. Pelegri-Llopart et al. teach creation of an interface specific stub at compile time. Figure 6 blocks 492 and 496 show extraction or introspection of the remote or target interfaces for stub generation. See column 8 lines 11 et seq. for elaboration.].

8. Claims 13 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hill et al. (US Patent 5,724,588) as modified by Object Management Group (“The Common Object Request Broker: Architecture and Specification”), Stutz et al. (US Patent 5,517,645), Hughes (“JavaBeans and ActiveX go head to head”), and as further modified by Hamilton et al. (US Patent 5,737,607) as applied to claim 11 above, and further in view of Hollberg et al. (EP 0 727 739 A1)

As to **claim 13**, Hill et al. as modified by OMG for claim 11 teaches the limitations substantially as claimed except the target object is a managed object [Hollberg et al. teaches agent based management of managed objects. There is disclosed both ProxyAgent Objects in column 10, as well as Proxy Managed Objects in column 18 as local representatives of remote entities, see abstract. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the managed objects as disclosed by Hollberg et al. into the distributed CORBA based

Art Unit: 2151

framework of Hill et al. as modified, because Hollberg recognized that “the object-oriented interface (OOI) for the use in OSI management applications and the related Object Interface Composer (OIC), minimize the effort needed to build the communication related functions of management applications.”] and the client application is a network management application [Hollberg et al. disclose an interface for converting network management application programs into network communication protocols, see abstract.].

As to **claim 14**, Hill et al. as modified by OMG and further modified by Hollberg et al. for claims 13 teaches the limitations substantially as claimed except the remote access support mechanism comprising a software mechanism [It would have been obvious to those of ordinary skill in the art at the time the invention was made to recognize the equivalency and interchangeability between software “entities” and hardware mechanisms.].

Conclusion

Any inquiry concerning this communication should be directed to **Gary Fourson** at telephone number **(703) 305-4392** or E-mail at the address **gary.fourson@uspto.gov**.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is **(703) 305-3900**.

The fax numbers for Official (703-746-7239), to be intended for entry into the application, Non-Official/Draft (703-746-7240), or After-final (703-746-7238) communications may be utilized for expedited transactions.

gsf

January 4, 2002